Jurnal Ilmu Komputer dan Informasi (Journal of Computer Science and Information)
12/2 (2019), 91-102. DOI: http://dx:doi:org/10:21609/jiki:v12i2:745

LEARNING WORD RELATEDNESS OVER TIME FOR TEMPORAL RANKING

Dinda Sigmawaty and Mirna Adriani

Faculty of Computer Science, Universitas Indonesia, Kampus UI, Depok, 16424, Indonesia

E-mail: dinda.sigmawaty@ui.ac.id, mirna@cs.ui.ac.id

Abstract

Queries and ranking with temporal aspects gain significant attention in the field of Information Retrieval. While searching for articles published over time, the relevant documents usually occur in certain temporal patterns. Given a query that is implicitly time-sensitive, we develop a temporal ranking using the important times of query by drawing from the distribution of query trend relatedness over time. We also combine the model with Dual Embedding Space Model (DESM) in the temporal model according to document timestamp. We apply our model using three temporal word embeddings algorithms to learn relatedness of words from news archive in Bahasa Indonesia: (1) QT-W2V-Rank using Word2Vec (2) QT-OW2V-Rank using OrthoTrans-Word2Vec (3) QT-DBE-Rank using Dynamic Bernoulli Embeddings. The highest score was achieved with static word embeddings learned separately over time, called QT-W2V-Rank, which is 66% in average precision and 68% in early precision. Furthermore, studies of different characteristics of temporal topics showed that QT-W2V-Rank is also more effective in capturing temporal patterns such as spikes, periodicity, and seasonality than the baselines.

Keywords: *Information Retrieval, temporal ranking, Dual Embedding Space Model, temporal word embeddings*

Abstrak

Kueri dan pemeringkatan dokumen dengan aspek temporal memiliki perhatian yang signifikan dalam bidang Perolehan Informasi. Saat mencari artikel yang diterbitkan dalam periode waktu yang panjang, dokumen relevan biasanya muncul dalam pola tertentu. Diberikan sebuah kueri yang implisit dan sensitif terhadap waktu, kami mengembangkan teknik pemeringkatan temporal menggunakan waktu-waktu penting kueri yang diperoleh dari distribusi keterkaitan kata dari waktu ke waktu. Kami juga menggabungkan model Dual Embedding Space Model (DESM) yang dibangun dengan teknik temporal sesuai dengan waktu pembuatan dokumen. Kami menerapkan model kami menggunakan tiga algoritma *temporal word embeddings* untuk mempelajari keterkaitan kata dari arsip berita dalam Bahasa Indonesia: (1) QT-W2V-Rank menggunakan Word2Vec (2) QT-OW2V-Rank menggunakan OrthoTrans-Word2Vec (3) QT-DBE-Rank menggunakan Dynamic Bernoulli Embeddings (DBE). Skor tertinggi dicapai dengan Word2Vec yang dipelajari secara terpisah dari waktu ke waktu, yang disebut QT-W2V-Rank, yaitu 66% dalam presisi rata-rata dan 68% pada presisi awal. Teknik yang diusulkan juga diuji pada beberapa topik temporal yang memiliki pola berbeda, hasilnya menunjukkan bahwa QT-W2V-Rank lebih efektif dalam menangkap kueri yang memiliki pola seperti tren, periodisitas, dan musiman daripada penelitian sebelumnya.

Kata Kunci: *Perolehan Informasi, pemeringkatan temporal, Dual Embedding Space Model, temporal word embeddings*

1. Introduction

Documents such as internet archives, news, and twitter feeds have topics which are constantly evolving and being replaced. Thus, time becomes an important concept in retrieving these documents. There are a growing number of both corpora and individual users that require documents that are not only topically relevant but also created during the most relevant time periods. Metzler et al. [1] report that almost 1.5% of queries contained an explicit time and 7% contained an implicit time. Other studies by Zhang et al. [2] have shown that 13.8% of queries contained an explicit time and 17.1% of queries contained an implicit time. Explicit queries have clear time information and can immediately position time without further knowledge, for

example, the "2018 Asian Games", and "Jakarta 1990". In contrast, implicit queries do not define time directly, such as: "Habibie's Presidency", or "plane crash MH17". In this type of query, more knowledge is needed to gain the important time of queries. For example, the query "Habibie's presidency" requires a process to find out the exact period time when Habibie was president. Users are more likely to write queries in implicit types, so it is difficult to position the concept of time and understand the intent of the user [2].

Temporal Information Retrieval (T-IR) is related to user querying behavior that might vary over time and present certain temporal patterns, such as, spikes, periodicity, and seasonality. When ranking documents in TIR, they should be ranked higher if their creation dates closely matches the time of the queries. One successful approach on temporal ranking requires an initial retrieval system purposed by Campos et al. [17]. They extracted dates from top-n web snippets to determine the time distribution of the relevant documents. Rao et al. [3] explored an alternative approach, called query trends, which uses the temporal statistics of the query terms in the collection to indicate relevance. Another work in ranking uses word embedding, called the Dual Embedding Space Model (DESM), is proposed by Mitra et al. [8]. This technique considers the relationship of query terms with all the words in the document but does not pay attention to the time aspect.

Campos et al. [18] provided a general overview of T-IR systems as well as a number of promising research directions, one of which is Temporal Text Similarity. Capturing temporal text similarity or relatedness has many interesting challenges in the fields of Natural Language Processing (NLP) and Information Retrieval (IR). In NLP, it aims to learn to capture time-sensitive meanings of words. For example, "apple," which was previously only associated with fruit, is now also associated with a technology company. For achieving this goal, many researchers have used temporal word embeddings. Recently, Kim et al. [19] computed static word embeddings in each time slice separately without performing smoothing to make the embeddings comparable across time. Hamilton et al. [21] have found a way to align the word embeddings across time slices to ensure that the vectors are aligned to the same coordinate axes by imposing an orthogonal transformation after word embeddings were trained. Finally, Rudolph and Blei [6] propose a model to learn word embeddings across time jointly without training it separately. In IR, a temporal word embeddings algorithm can be used to improve the effectiveness of web archive searches. Rosin et al. [4] have employed an algorithm that learned word relatedness over time by understanding user query intent influencing query expansion. They modeled relatedness change over time as a time series to their classification system. Given two entities, the system will predict whether they relate to each other during a referenced year, and the related entities are then reformulated by expanding the user query.

Previous work on temporal ranking used initial retrieval and query trend frequency to gain more knowledge about user query intent. In this work, we apply the temporal word embeddings method to learn query trend relatedness over time influencing document ranking. The essential method is to build the query trend using several temporal word embeddings algorithms to measure word relatedness over time. We are motivated by the query trends hypothesis, [3] which stipulates that there is a correlation between query trends and the distribution of relevant documents. We describe several models using these temporal embeddings. The model is further compared for advantages in capturing query user intent and classification of the important and non-important time references. We also combine the model with DESM in the temporal model according to document timestamps. We present an evaluation of our approach using several trials to rank algorithms and a comparison with the lexical and temporal baseline ranking models. Finally, we analyze the positive aspects of our method according to different characteristics of temporal query topics.

The remainder of this paper is structured as follows: in section 2 we present the background and related work; in section 3 we introduce our proposed method; experimental setups and results are discussed in Section 4 and Section 5; finally, we conclude this paper in section 6 with several analyses.

2. Background and Related Works

2.1 Ranking

One of the ranking techniques used in Information Retrieval is the Dual Embeddings Space Model (DESM) by Mitra et al. [8]. The technique considers the relationship of query terms with all words in the document. Word relatedness is trained using Word2Vec with Continuous Bag of Words (CBOW) and Negative Sampling. They found that CBOW can model the "aboutness" of a document by mapping the input and output vectors. The cosine similarity in input-

output vectors tend to have a higher score between words that often have co-occurrence in training data, and it can represent the relatedness of words. They have built a document ranking model as follows [8]:

$$DESM(Q,D) = \frac{1}{|Q|}\sum_{q_i \in Q} \frac{q_i^T \bar{D}}{\|q_i\| \, \|\bar{D}\|} \quad (1)$$

Where $\bar{D}$ is the centroid of all the normalized vectors for the words in the document, with the formula [8]:

$$\bar{D} = \frac{1}{|D|}\sum_{d_j \in D} \frac{d_j}{\|d_j\|} \quad (2)$$

The results of this study outperform LSA and BM25 in document ranking. Unfortunately, DESM does not pay attention to the temporal aspect.

The ranking techniques for temporal queries have document targets within different time periods, so the ranking techniques are also different from a classical information retrieval system. Intuitively, documents more relevant and have higher ranking score if the creation dates closely match with the time of queries [11]. Ranking techniques that integrate temporal aspects proposed by Rao et al., [3] utilize topical and temporal features. Topical features are captured using query likelihood to help determine similarity between document and query. Temporal features include those based on the query trend, which is the relative entropy of the representative unigram and bigram query terms, and a density estimation of the document's timestamps of the representative unigram and bigram. Another work by Campos et al. [17] proposed a linear combination of topical and temporal scores extracted within n-top web snippets called GTE-Rank. From web snippets, they extracted relevant words/multi-words and dates. The temporal similarity measurement is called GTE, which evaluates the degree of relation between candidate date and query. GTE uses InfoSimba (IS), a vector space model supported by corpus-based token correlation based on its frequency and inverse document frequency. IS calculates the correlation between word-only context vectors, date-only context vectors and the combination of words and dates.

2.2 Temporal Modelling of Pseudo Trends

To find documents that are temporally relevant to the given queries, an information retrieval system must be able to know the distribution of relevant documents over all times. The relevance of documents can be estimated through their temporal distribution from the initial retrieved documents. For example, Campos et al. [17] have used n-top web snippets to extract relevant words/multi-words and dates. In contrast, many researchers try to exclude initial retrieval to make the system faster and to overcome dependence on the efficiency of the system used for the initial retrieval. Therefore, they build query trends to find where the queries are mentioned most frequently to indicate the interval of document relevance. For example, for the query "Governor Joko Widodo," the relevant documents are articles about Joko Widodo's role as Jakarta's Governor. Intuitively, the most frequent occurence the query terms "Governor Joko Widodo" should be in the few years are also when the most relevant documents are clustered.

Asur and Buehrer [24] build query trends using query clicks over time to understand the temporal pattern of the query. The trends are then used to classify queries as navigational, adult, or news. Ren et al [25] have also classified query trends using frequency distribution over time in web query logs. Another work by Costa et al. [20] identifies query trends by exploiting the variance of web characteristics over time. Their hypothesis is that the more similar the web characteristics, the closer the periods of documents are. Finally, Rao et al. [3] use the statistical distribution of query terms represented by unigrams and bigrams based on their occurrence in the document collection. This approach tries to find more relevant documents in temporal intervals where the query terms tend to appear in bursts.

2.3 Temporal Text Relatedness

2.3.1 Word Embeddings

Word embeddings aim to represent words with low-dimensional vectors where those with similar context are closer in semantic space. The techniques first used in the 90s relied on a statistical approach [22]. Recently, some computational advances with neural networks have been proposed, such as Word2Vec [7, 9] and Dynamic Bernoulli Embedding [6], that has improved the performance of word representation greatly. Word embeddings also can be used to represent relatedness over time and to understand relatedness extracted from document collection [4].

2.3.2 Word2Vec

Word2Vec is a word embedding technique that can model words in vector space using a neural network architecture [12]. This technique was first

proposed by Mikolov et al. in 2013 [7] and has been widely used and developed for representing words. In Word2Vec, words are represented in their environment (neighboring words), which is called the word context. The idea is that if "word A" and "word B" have identical environments, then the words are in a similar context. There are different ways of generating the inputs and the expected outputs to the neural network. One of these methods is called Continuous Bag-Of-Word (CBOW). This method essentially takes a word as a target and then uses word context to predict the word.

One optimization technique on Word2Vec is negative sampling [9]. Negative sampling aims to solve problems in large datasets. In updating rules, it is very inefficient to train all vectors and update all output vectors. However, with negative sampling, instead of updating all output vectors, we can do the sampling. The sampling is a negative sample of each word and therefore the model only updates some output vectors that are negative samples from input vectors [12].

2.3.3 Temporal Word Embeddings

The field of Natural Language Processing (NLP) is an active research topic in understanding changes in the meaning of a word, which is called word evolution. In order to compare word vectors from different time-periods, a previous study by Rosin et al. [4] used Word2Vec running separately at all times. More recently, Hamilton and Jurafsky [21] use the orthogonal Procrustes problem to align the learned low-dimensional embeddings. To do this, they propose a two-step procedure: first, they learn word embeddings $\boldsymbol{Y^{(t)}}$ each year t separately and afterward solving an orthogonal Procrustes problem between $\boldsymbol{Y^{(t)}}$ and $\boldsymbol{Y^{(t+1)}}$.

Other researchers have modified word embedding vectors to share time information using a latent diffusion process [5] and random walk [6]. Their models can capture the evolution of words better than Word2Vec. Rudolph and Blei [6] report that their word embeddings not only capture changes in the meaning of words over time but also changes in the dominant meaning and the relevance of the subject. The model they propose is called Dynamic Bernoulli Embeddings (DBE). DBEs generalize CBOW and infer the embedding based on a Stochastic Gradient Descent (SGD) and connect to negative sampling. The neural network uses a prior distribution which is Gaussian with diagonal variance on the weight. Weighting on this model is based on the embeddings vector and context vector. Context vectors are shared across all positions in the text, but the embedding vectors are only shared within a time slice. The probabilistic perspective, the priors, and the parameter sharing allow the models to extend this setting to capture dynamics.

3. Proposed Method

The overall idea of the process is to identify time of query using query trend built by temporal word embeddings and classify years which are important for a given query to enhance the effectiveness of temporal ranking on four different steps showed in Figure 1. We explain query trends processing, date classification, temporal similarity, and ranking in the remainder of this section.

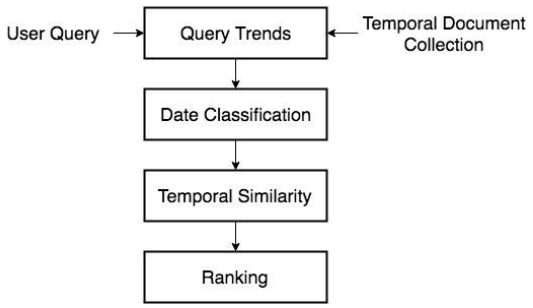

Figure. 1. Overall architecture

3.1 Query Trend Processing

Given a query, we built query trends by computing word relatedness on a bigram and skipgram of query terms from the document collection. We need to precompute word relatedness over time for the entire year in the collection to build the query trends feature. Relatedness was captured by using word embeddings that were generated for every year period. We used Word2Vec (W2V) [9], OrthoTrans-Word2Vec (OW2V) [21] and the Dynamic Bernoulli Embeddings (DBE) [6] model. We performed an exploration of different configuration for the trained word embeddings model. The Word2Vec models had the following parameters: window size was 12, dimensionality was 200, and negative sampling was 10. For DBE, the best log-likelihood score was the model with these parameters: an embedding size of 200, a window size of 4, and a negative sample of 10.

Our query trends are illustrated in figure 2, which shows the distribution of query trends to relevant documents using the query term "*Gubernur Joko Widodo*" (Governor Joko Widodo). As can be seen, there is a strong correlation between the query trends (especially

for the bigram "*gubernur joko*" and the skipgram "*gubernur widodo*") with the ground truth of relevant documents. The distributions that did not show a lot of spikes or are nearly uniform ("joko widodo") were usually less useful. Spikes from bigram and skipgram distributions on queries were measured using entropy. The smaller the entropy value indicates that the distribution was getting away from uniform. For example, given one distribution $t = \{c_1, c_2, \dots, c_n\}$ from bigram or skipgram query at n time, then entropy value can be calculated by [28]:

$$Entropy(t) = -\sum_{i=1}^{n} \frac{c_i}{\sum_i c_i} \log \frac{c_i}{\sum_i c_i} \quad (3)$$

Where c is cosine similarity between two words in the distribution. The higher the distance score between two words at one time, the more relevant the word is. The distance score is calculated by cosine similarity.

We selected bigram and skipgram distribution with the lowest entropy as the representative bigram and skipgram query trends. The output of this step is cosine similarity of representative bigram and skipgram query trends over time.

$$\text{BigramRepresentative}(q; v_i, v_{i+1}) = \langle \cos(v_i^{t_1}, v_{i+1}^{t_1}), \dots, \cos(v_i^{t_n}, v_{i+1}^{t_n}) \rangle \quad (4)$$

$$\text{SkipgramRepresentative}(q; v_i, v_{i+2}) = \langle \cos(v_i^{t_1}, v_{i+2}^{t_1}), \dots, \cos(v_i^{t_n}, v_{i+2}^{t_n}) \rangle \quad (5)$$

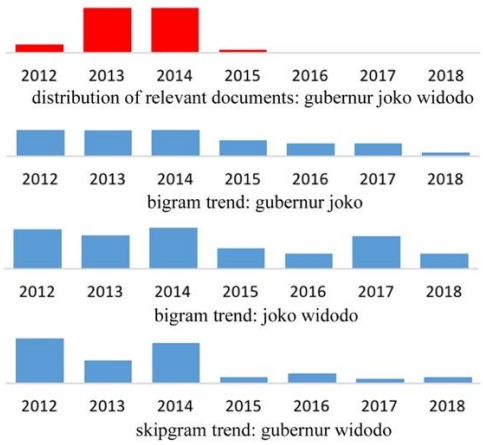

Figure. 2. Ground truth of relevant documents (red) and query trend distribution (blue) from "*gubernur joko Widodo*" (governor joko widodo)

3.2 Date Classification

This methodology aims to determine the important years from query trends. The query years are first classified with regard to their peak before being sent to the ranker. We applied the peak detection algorithm by Rosin et al. [4] to remove non-important dates because the algorithm was capable of detecting peaks and periods of continuity. Given a representative bigram and skipgram query trends, the system finds the local maximum for each pair and added to a list of the important date showed in Figure 3.

```
Peak Detection Algorithm :
Given Representative bigram Rb,
Representative skipgram Rs,
AbsoluteThreshold, RelativeThreshold,
PlateauThreshold
1.  Qt ← merge(Rb, Rs)
2.  Peaks ← Find the relative extrema of Qt
3.  MaxPeak ← Find the max value in Qt
4.  Peaks ← [year, value] in Peaks if value
    < AbsoluteThreshold and value <
    (RelativeThreshold  * MaxPeak)
5.  For each [year, value] in Peaks
6.     If abs(Qt [year, value]/Qt[year-1,
       value]) -1 < PlateauThreshold then
7.        Peaks ← append Qt[year-1, value]
8.     If abs(Qt [year, value]/Qt[year+1,
       value])-1 < PlateauThreshold then
9.        Peaks ← append Qt[year+1, value]
Output: Peaks
```

Figure 3. Peak detection for Date Classification

First, we have to merge the Representative bigram (Rb) and Representative bigram (Rs) for Query Trend (Qt) which only using one distribution. The merged used SUM or MAX function, where SUM obtained from the sum of Rb and Rs, and MAX is selecting one of the distributions that have minimum entropy value. Furthermore, we compare both functions which could give the best result. The second and third line is finding the relative extrema and maximum value of Qt. Some thresholds were used; the first is an absolute threshold to filtering out the peaks that were not relevant if they are below the value, while the second, called relative threshold, removes points that are much lower than the highest maximum. The filtering process using the two thresholds is applied on line 4. Line 5-9 presented for each peak point candidate, the algorithm compares the points of the surrounding neighbors using a plateau threshold. If the value is close to the current peak, then it is added to list of important dates, otherwise, it is not.

3.3 Temporal Similarity

This technique uses a probabilistic approach of words in a document to identify relevance. The appearance of the word being considered is the result of the word embeddings query from all the words in the document. We used the output and

input vectors from our pre-trained word embeddings. In Word2Vec we had some input vector and some output vector but in DBE we had just one input vector and a few output vectors. First, we calculated the centroid of query terms in all queries using input vectors, which we called the query vectors. Word2Vec could perform differently each year, whereas DBE was constant for each year. Second, we calculated cosine similarity from query vector to every word in the document using output vectors corresponding to the document timestamp. For example, suppose there were a query, Q, for document D1 written in 2016 and document D2 written in 2015. The DESM value from Q-D1 was calculated by query vector and vector out in 2016. Likewise, the documents that were written in the year 2015 (Q-D2) used the output vector in 2015. In other words, we perform DESM to capture a dynamic called the Temporal Dual Embedding Space Model (TDESM) according to the word embeddings of the document timestamp.

3.4 Ranking Algorithm

Many learning-to-rank algorithms have been proposed in the literature. Among of these approaches, we employed the following three ranking algorithms based on three approaches, which are pointwise, pairwise, and listwise [16]:

- **LinearRegression:** Linear regression (pointwise) contained a feature vector of each single document. The algorithm was modeled as a regression that takes the feature vector of a document as input and predicts the relevance degree of document [16].
- **RankNet:** RankNet (pairwise) was developed using a neural network and optimizes the loss function using Stochastic Gradient Descent. The loss function aims to minimize the incorrect order among a pair of result. Given a set of pairs of samples [A, B] in $\mathcal{R}^d$ together with target probabilities $\bar{P}_{AB}$, the algorithm learns that sample A is to be ranked higher than sample B [15].
- **Coordinate Ascent:** The coordinate ascent (listwise) algorithm proposed by Metzler and Croft [10] finds a parameter setting for the best value to map the features of the query and document pairs. To find best values for parameters, this technique needs a set of training data T and an evaluation function E($R_w; T$). Moreover,

$$\hat{w} = argmax_w \, \mathrm{E}(R_w; T) \quad (6)$$

Where $\boldsymbol{R_w}$ is the set of rankings produced by the scoring function for all the queries. The goal of this model is to find a parameter setting that maximizes E for the training data T**.**

4. Experimental Setup

4.1 Dataset Construction

For constructing the corpora, we used articles from several Sindonews[1] portal (national[2], international[3], metro[4], sports[5]) between 2012 and 2018 collected from Internet Archive[6]. News corpora offer natural advantages for studying trends and have larger knowledge bases. The general statistics are detailed in Table 1. We extracted title, link, date, and content for every article and performed preprocessing strategies like cleansing from non-alpha numeric character, stemming and stopword removal. Subsample data for training and testing selected using the open source implementation of the retrieval system of Lemur Project called Indri Query Language[7]. We then retrieved 100 articles for each 15 temporal queries.

TABLE 1
DOCUMENT COLLECTION STATISTICS

Crawler	Internet Archive
Size of documents	303143
Date range	2012 to 2018
Average document length	251
Size of temporal queries	15
Average query length	3.13
Size of assessed document	1500

4.2 Evaluation Methodology and Metrics

To gather relevance judgments we use the Cranfield paradigm [26]. We provided three annotators with a two-point scale of relevance judgments: relevant and not relevant. We follow manual assessment in [27] to presenting guidelines and articles for annotator. The inter-agreement between judges measured by Fleiss's kappa and obtained 0.45 which can be seen as a moderate agreement.

We measure quality of the returned list using Mean Average Precision (MAP), Precision at 30

[1] https://sindonews.com
[2] https://nasional.sindonews.com
[3] https://international.sindonews.com
[4] https://metro.sindonews.com
[5] https://sports.sindonews.com
[6] https://web.archive.org
[7] https://sourceforge.net/projects/lemur

(P30) and Precision at 10 (P10). We used cross-validation with 5-fold to perform validation. We also analyze our model in different temporal query characteristics to gain more understanding about the effectiveness.

4.3 Ranking Features and Models

The following features are proposed in this work:

- **TDESM** features estimate the similarity between the query and documents to capture topical and temporal similarity.
- **Relative entropy from bigram.** Representative bigram query term distributions are calculated as described in 3. The Relative entropy is computed as the absolute difference between the bigram entropy and the maximum entropy, where the maximum entropy is uniform distribution.
- **Relative entropy from skipgram.** This feature is exactly the same as that described above, except that it uses a skipgram.
- **Word relatedness score from bigram.** This is the distance between query terms in bigram representation with cosine similarity and pre-computed word embeddings.
- **Word relatedness score from skipgram.** This feature is exactly the same as described above, except that it uses a skipgram.

The query years were first classified with regard to their peak before being sent to the ranker. We filtered out the set of all non-relevant date from input of the temporal similarity measure. This allows filtering according to relative entropy and word relatedness.

We would like to understand the contribution of each word representation models using these features. The description of different approaches is given as follows.

- **QT-TF-Rank** using Term Frequency for building query trends and date classification. The features used are Query Likelihood (QL), relative entropy from the representative unigram and bigram (with frequency signal), and the density estimation from the representative unigram and bigram.
- **QT-W2V-Rank** using features such as Temporal Dual Embedding Space Model (TDESM) with Word2Vec, Relative entropy from representative bigram and skipgram (with relatedness signal using Word2Vec), and Word relatedness score from the representative bigram and skipgram (with relatedness signal using Word2Vec).
- **QT-OW2V-Rank.** This method used the feature in exactly the same way as described for QT-W2V-Rank, except that it computes the TDESM and relatedness signal using OrthoTrans-Word2Vec (OW2V).
- **QT-DBE-Rank.** This method uses the feature in exactly the same as described on QT-W2V-Rank, except that it builds on word embedding to calculate word relatedness and TDESM using Dynamic Bernoulli Embeddings (DBE).

5. Experimental Result

5.1 Baselines

Five baselines were used as points of comparison. Query Likelihood (QL) [13] and the Dual Embedding Space Model (DESM) [8] were used for ranking baseline with topical relevance, GTE-Class [17] were used as temporal classification baseline. For temporal ranking baseline, we used QT [3] and GTE-Rank [17].

- **Query likelihood** approach of Ponte and Croft [13]. This approach using language modeling framework. Documents are ranked by $P(D|Q) \propto P(Q|D)P(D)$, where $P(Q|D)$ represents the likelihood that the language model that generated document D would also generate query Q. $P(Q|D)$ was the posterior and $P(D)$ was the prior distribution.
- **Dual Embedding Space Model (DESM)** of Mitra et al. [8]. We used Word2Vec with CBOW and negative sampling, and also in and out vectors to capture the relatedness of words.
- **Query Trend Frequency (QT)** of Rao et al. [3]. This method uses Query Likelihood [13], relative entropy from representative unigram and bigram distribution (with the term frequency signal), and the density of the document in unigram and bigram query terms (also with term frequency signal) as features.
- **GTE-Class** of Campos et al. [17] proposes a technique to determine whether the year is relevant or not for user a query using a candidate year obtained in the document content. This technique performs classification based on the threshold strategy. We used their provided web services[1] to perform classification and obtain $\lambda = 0.35$ for best classification configuration.
- **GTE-Rank** of Campos et al. [17] extract important dates and keywords from n-top web snippets from a given query (in our work the

value of n is 30). We used their web services[8] to perform extraction and then calculated IS and GTE score as described in [17, 23].

5.2 Result

Table 2, 3, and 4 summarized the result of our experiments.

TABLE 2
AVERAGE PRECISION (AP) AND AVERAGE RECALL (AR) ON DATE CLASSIFICATION

Method	AP	AR
QT-TF-Class	0.76	0.60
QT-W2V-Class	0.85	0.70
QT-OW2V-Class	0.73	0.53
QT-DBE-Class	0.44	0.80
GTE-Class [17]	**0.83**	**0.87**

Candidate dates were extracted based on the rule-based model with peak detection algorithm, each query and date pair was then manually labeled. Table 2 showed result on data classification method using five approaches: QT-TF-Class, QT-W2V-Class, QT-OW2V-Class, and QT-DBE-Class, along with GTE-Class as the baseline. For capturing representative bigram and skipgram distribution, QT-TF-Class using Term Frequency, QT-W2V-Class using Word2Vec, QT-OW2V-Class using OrthoTrans-Word2Vec and QT-DBE-Class using DBE were applied. The models evaluated used Average Precision (AP) and Average Recall (AR). In order to determine the best value of the threshold, we performed some heuristic experiments and obtained the best value of absolute threshold = 0.1, plateau threshold = 0.2, and relative threshold = 0.6. From table 1, we can observe that GTE-Class achieved the best performance in terms of AP, which is 0.83 and 0.87 in AR. In this task, our model failed to outperform the baseline model. QT-DBE-Class was consistently the worst performer, while QT-W2V-Class outperformed the rest. We found that DBE could not provide good results when the time span was not long enough. This study used the years 2012-2018, which was the signal of word relatedness given by DBE and was almost unchanged due to the short time vulnerability. The distributions therefore appeared almost uniform in each query. The model was made to capture word dynamics smoothly and the output vector controls the input vector in the current year, ensuring that it is not far away from the previous year. QT-W2V-Class also outperformed the QT-OW2V-Class that aligned the learned low-dimensional embeddings to capture dynamic word comparisons. Word2Vec performed better on this task because it did not have that capability. The distribution did not consider the value of the previous years, Instead, words were learned independently over each year. Consequently, the relatedness of the distribution word became more uneven and appeared in bursts. It therefore had a higher relative entropy value and was more useful in capturing bigram and skipgram word relatedness.

Our next step was to validate our temporal ranking model. Table 2 shows our ranking model against QL, the original model DESM, QT, and GTE-Rank. In our models, QT-W2V-Rank using QT-W2V-Class, QT-OW2V-Rank using QT-OW2V-Class, and QT-DBE-Rank using QT-DBE-Class for date classification. The results show that QT-W2V-Rank achieves the best results, both in improving early precision and in average precision. The model can outperform temporal baselines using word frequency (QT), meaning that the relatedness signal is more useful than frequency. In our experiment, word frequency failed to distinguish between Anies Baswedan serving as a minister and as a governor. They placed both at the same time, while our model can distinguish between the two points of their career. QT-W2V-Rank consistently outperformed QL and DESM. This suggests that models with temporal signals are more effective than models that only use a lexical signal for relevance ranking. QT-W2V-Rank also outperformed models that used initial retrieval in ranking (GTE-Rank).

[8] http://www.ccc.ipt.pt/~ricardo/software.html

TABLE 3
OUR MODEL AGAINST TEMPORAL RANKING BASELINE IN DIFFERENT RANKING ALGORITHMS

Method	RankNet			Coordinate Ascent			Linear Regression		
	MAP	P10	P30	MAP	P10	P30	MAP	P10	P30
QL [13]	0.53	0.48	0.51	0.55	0.49	0.53	0.56	0.52	0.55
DESM [8]	0.59	0.59	0.55	0.60	0.63	0.58	0.60	0.62	0.58
QT [3]	0.54	0.51	0.52	0.57	0.55	0.55	0.56	0.54	0.53
GTE-Rank [17]	0.56	0.55	0.55	0.58	0.58	0.58	0.58	0.57	0.55
QT-W2V-Rank	0.66	0.68	0.62	**0.66**	**0.68**	**0.67**	0.64	0.66	0.63
QT-OW2V-Rank	0.59	0.58	0.57	0.61	0.63	0.58	0.60	0.62	0.58
QT-DBE-Rank	0.52	0.50	0.50	0.53	0.54	0.51	0.56	0.57	0.54

Average	**0.57**	**0.55**	**0.54**	**0.58**	**0.58**	**0.57**	**0.58**	**0.58**	**0.56**

TABLE 4
RESULT ON EACH QUERY CHARACTERISTIC

Method	Person Entities			Specific Event			General/Periodic Event		
	MAP	P10	P30	MAP	P10	P30	MAP	P10	P30
QT [3]	0.40	0.33	0.35	0.81	0.82	0.82	0.46	0.42	0.43
GTE-Rank [17]	**0.47**	**0.46**	**0.46**	0.82	0.85	0.90	0.44	0.42	0.37
QT-W2V-Rank	0.45	0.42	0.38	**0.84**	**0.89**	**0.90**	**0.64**	**0.67**	**0.75**
QT-OW2V-Rank	0.43	0.36	0.39	0.81	0.85	0.80	0.58	0.54	0.62
QT-DBE-Rank	0.35	0.32	0.37	0.75	0.75	0.73	0.48	0.45	0.57

Our experiment showed that Coordinate Ascent algorithm surpassed RankNet and Linear Regression on average. This suggests that a listwise approach to the set of documents associated with the query to predict the correct order of documents was most capable in identifying the relevant documents using our features. For complete assessment of the overall effectiveness of our purpose model, we performed 30 random split experiments for all temporal ranking models and baselines using the Coordinate Ascent algorithm. The summary of data is shown in a box-and-whiskers plots in Figure 4. Each box shows the spread and center of the data, where at the ends of the box is first quartile and the third quartile. At the end of the bottom whisker is the minimum number in the data, whereas the far up is the maximum value. The horizontal line in the center of the box represents the median value. Figure 4 provides more evidence for the effectiveness of our purpose model. It is clear that QT-W2V-Rank was consistently more effective than other proposed models and the temporal baselines (QT and GTE-Rank). While our date classification method alone does not outperform the baseline, combining them with our temporal feature does yield a boost in effectiveness. This was shown in our experiments, where the most useful feature was Word2Vec for captured word relatedness and query trends.

5.3 Per-topic character Analysis

We break down our results into three query characteristics to show the effectiveness in our models in different temporal query pattern. We used our temporal model (QT-W2V-Rank, QT-OW2V-Rank, and QT-DBE-Rank) against the baseline (QT and GTE-Rank) as shown in table 4. The first was person or entity: this query used to identify personal names in topics like "*Gubernur Joko Widodo*" (Governor Joko Widodo). Relevant documents were documents when Joko Widodo served as governor, and it usually in a given time span. The second was Specific Event: this query described something that happened at specific or one time. The example was "*Bom Sarinah atau Thamrin*" (Sarinah or Thamrin bombs), incidents that occurred in 2016. The third was the general event query, which is used when the topic refers to more than one specific time. This can be a periodic event such as "Asian Games" or not periodic but occurring at more than one time such as "*Perombakan Kabinet Joko Widodo*" (Joko Widodo's Cabinet reshuffle).

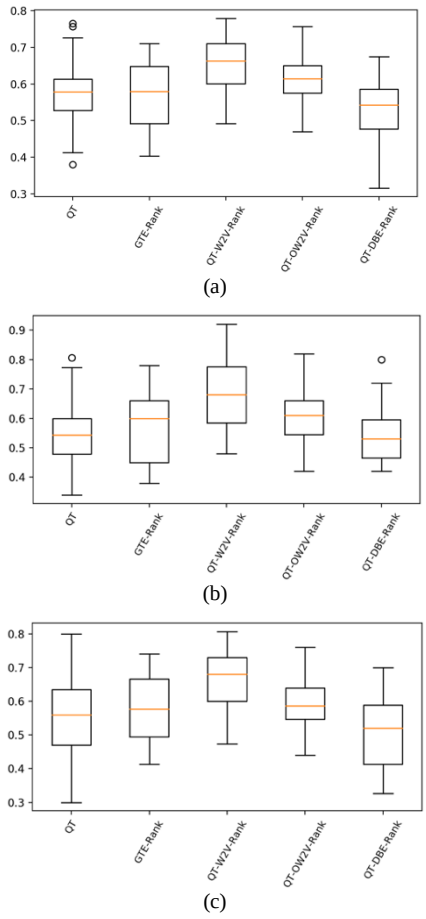

Figure 4. Box-and-whiskers plots summarizing how much each temporal model outperforms QT and GTE-Rank

baselines across 30 random trials in terms (a) MAP, (b) P10, and (c) P30

We started our investigation from general event queries. First, our model QT-W2V-Rank performed best in average and early precision. This shows that the relatedness signal captured by Word2Vec can be useful in queries where the relevant documents are periodic or occur more than once. Second, for specific event queries, we can see from table 4 that almost every method could give a good result. This is obviously because specific event queries had low ambiguity — the topic happened at one specific time, and never happened at another time. Our model QT-Class-W2V performs best in terms of MAP, P30, and P10 with 0.84, 0.89, and 0.90 scores respectively. This shows that our model is also capable of capturing relevant documents for queries for events that occur at one time, specific or sharp.

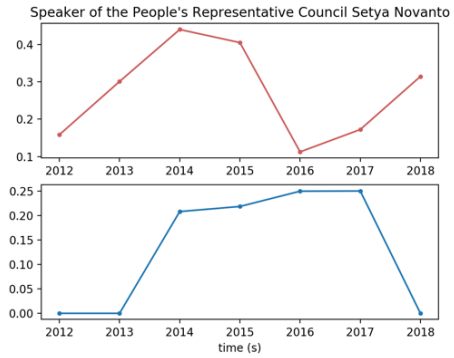

Figure 5. Sum of bigram and skipgram representative distribution (red) and distribution of relevant document (blue) on query "Ketua DPR Setya Novanto" (Speaker of the people's Representative Council Setya Novanto) on QT-W2V-Rank

Third, for person or entity queries, GTE-Rank outperforms all others at early and average precision. We can observe that the years extracted from the contents of the documents are more accurate guide than using the creation dates for time span query or identifying personal names in topics. Also, our model failed to hold relevance when queries occur over longer periods of time. For example, for the query "*Ketua DPR Setya Novanto*" (Speaker of the people's Representative Council Setya Novanto) presented on Figure 5., the figure shows that the sum of bigram and skipgram representation (red) peaks in 2014 and 2015. The peaks not relate to the distribution of relevant documents (blue), which have wider peaks from 2014 until 2017. The unbalanced data across the years may lead to miscaptured relatedness for person entities. However, GTE-Rank failed to capture periodic and specific events better than our approach.

6. Conclusion

This work contributes to a long thread of research on exploiting temporal signals for relevance ranking. While our date classification method fails to outperform the baseline, combining it with query trends relatedness and Temporal DESM yields clear improvement over the initial retrieval and frequency-based approach. Our approach (QT-W2V-Rank and QT-OW2V-Rank) outperforms all non-temporal and temporal baselines under most conditions. We can conclude that the relevance of words built by word relatedness over time with Word2Vec is useful in temporal ranking, and aligning the word embeddings to be comparable across time also produces a positive impact for query trend. DBE was not suitable for capturing relatedness in our corpus, and therefore it caused the QT-DBE-Rank model to fail to give a good result for temporal ranking. Our temporal model QT-W2V-Rank is also effective in capturing temporal patterns such as sharpness, periodicity, and seasonality, but not in the short time span given in this series of trials. In the future, we intend to test DBE performance using a longer time span.

References

[1] D. Metzler, R. Jones, F. Peng, & R. Zhang, "Improving search relevance for implicitly temporal queries" *In Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, ACM, pp. 700-701, 2009.

[2] R. Zhang, Y. Konda, & A. Dong, "Learning Recurrent Event Queries for Web Search" *In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Massachusetts, pp. 1129–1139, 2010.

[3] J. Rao, F. Ture, X. Niu, & J. Lin, "Mining the Temporal Statistics of Query Terms for Searching Social Media Posts" *In Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval*, ACM, pp. 133-140, 2017.

[4] G. D. Rosin, K. Radinsky, & E. Adar, "Learning Word Relatedness over Time" *In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 1168-1178, 2017.

[5] R. Bamler & S. Mandt, "Dynamic Word Embeddings" *In Proceedings of the 34th*

International Conference on Machine Learning, Sydney, pp. 380-389, 2017.

[6] M. Rudolph & D. Blei, "Dynamic Bernoulli Embeddings for Language Evolution," *In Proceedings of the 2018 World Wide Web Conference*, Lyon, pp. 1003-1011, 2018.

[7] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, & J. Dean, "Distributed Representations of Words and Phrases and their Compositionality" *In Neural Information Processing Systems 26 (NIPS)*, pp. 3111-3119, 2013.

[8] B. Mitra, E. Nalisnick, N. Craswell, & R. Caruana, "A Dual Embedding Space Model for Document Ranking," arXiv preprint arXiv:1602.01137, 2016. Extends WWW 2016 poster: Improving document ranking with dual word embeddings.

[9] T. Mikolov, K. Chen, G. Corrado, & J. Dean, "Efficient Estimation of Word Representations in Vector Space," *In Proceedings of the International Conference on Learning Representations*, Scottsdale, pp. 1-12, 2013.

[10] D. Metzler & W. B. Croft, "Linear feature-based models for information retrieval". *Information Retrieval 10(3)*, pp. 257-274, 2007.

[11] N. Kanhabua, & K. Nørvåg, "Determining time of queries for re-ranking search results" *In International Conference on Theory and Practice of Digital Libraries*, Springer, Berlin, pp. 261–272, 2010.

[12] X. Rong, 2014. "Word2Vec Parameter Learning Explained" arXiv preprint arXiv:1411.2738, 2014.

[13] J. M. Ponte & W. B. Croft, "A language modeling approach to information retrieval" *In Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, NY, Melbourne, Australia, pp. 275–281, 1998.

[14] R. Řehůřek, and P. Sojka, "Software Framework for Topic Modeling with Large Corpora" *In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, Valletta, Malta. ELRA, pp. 45–50, 2010.

[15] C. J. C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, & G. Hullender, "Learning to rank using gradient descent" *In Proceedings of the 22nd International Conference on Machine learning (ICML-05)*, pp. 89-96, 2005.

[16] T. Y. Liu, *Learning to rank for information retrieval*, Springer Science & Business Media, p. 267-268, 2011.

[17] R. Campos, G. Dias, A. Jorge, & C. Nunes, "GTE: a distributional second-order co-occurrence approach to improve the identification of top relevant dates in web snippets" *In Proceedings of the 21st ACM international conference on Information and knowledge management*, ACM, pp. 2035-2039, 2012.

[18] R. Campos, G. Dias, A. M. Jorge, and A. Jatowt. "Survey of temporal information retrieval and related applications". ACM Computing Surveys (CSUR) vol. 47(2), 2015.

[19] Y. Kim, Y. I. Chiu, K. Hanaki, D. Hegde, & S. Petrov, "Temporal analysis of language through neural language models" arXiv preprint arXiv:1405.3515, 2014.

[20] M. Costa, F. M. Couto, M. J. Silva, "Learning temporal-dependent ranking models" *In Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, ACM, pp. 757-766, 2014.

[21] W. L. Hamilton, J. Leskovec, & D. Jurafsky, "Diachronic word embeddings reveal statistical laws of semantic change" *In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 1489-1501, 2016.

[22] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, & R. Harshman, "Indexing by latent semantic analysis," *Journal of the American society for information science vol. 41(6)*, pp. 391-407, 1990.

[23] R. Campos. "Disambiguating Implicit Temporal Queries for Temporal Information Retrieval Applications," Ph.D Thesis, Department of Computer Science, Faculty of Sciences, University of Porto, 2013

[24] S. Asur & G. Buehrer, "Temporal analysis of web search query-click data" *In Proc. SNA-KDD*, Paris, France, ACM, pp. 1-8, 2009.

[25] P. Ren, Z. Chen, X. Song, B. Li, H. Yang, & J. Ma, "Understanding temporal intent of user query based on time-based query classification" *In Natural Language Processing and Chinese Computing*, Springer, Berlin, Heidelberg, pp. 334-345, 2013.

[26] D. Harman, *Information Retrieval Evaluation*. 1st ed, Morgan & Claypool Publishers, 2011.

[27] M. Costa & M. L. Silva, "Evaluating web archive search systems," *In International Conference on Web Information Systems Engineering*. Springer, Berlin, Heidelberg, pp. 440-454, 2012.

[28] S. Claude E, "A Mathematical Theory of Communication," *Bell System Technical Journal Vol. 27 Issue 3*, pp. 379–423, 1948.